4W-09

Minimum Diameter Multiple Steiner Tree Problem for Embedding Multiple VLANs

Andrew Lee[1,a)] Shuichi Miyazaki[2,b)] Yasuo Okabe[2,c)]

Abstract: Many network configuration design problems involve in finding optimal paths or trees within a given cost-constrained connected network. In designing VLAN, due to the lack of automation tool, most configurations are done manually, hence they are usually inefficient especially when the scale of the given network grows larger. In this paper we first define the criteria for evaluating the performance of a VLAN. Second, based on the criteria, we introduce a Minimum Diameter Multiple Steiner Tree (MDMST) problem as a mathematical model of the multiple VLAN configuration problem. We also implement heuristic procedures to solve the problem and evaluate the performance of the result using simulated data.

Keywords: VLAN, network configuration, Steiner tree, minimum diameter

1. Introduction

Many network configuration design problems involve in finding optimal paths or trees within a given cost-constrained connected network. One of the many reasons for why an optimal network configuration is crucial for system providers is that given the same amount of hardware resource and bandwidth limitation, it enables them to provide a much lower end-to-end latency service to their clients in compare to services with non-optimized configurations. For service providers, it often requires them to configure a network that is capable to support the construction of multiple Virtual Local Area Networks[1] (VLANs). However, even though extensively used, due to the lack of efficient automation tools to design multiple VLANs, most design methods up to now still requires these configurations to be done manually which can often lead to error-prone, time-consuming or inefficient results.

In this paper, we focus on designing an automation method to construct multiple VLANs in a given network with latency and bandwidth capacity as the main cost and constraint respectively to the problem that we propose to solve. We start by stating our problem formally in Section 2 and proceed to give the procedure to solve the problem in Section 3. We make an evaluation of our method in Section 4 and give a brief conclusion to our study in Section 5.

2. Problem Statement

We formulate the problem to find the best configuration of multiple VLANs with edge latency cost and edge bandwidth capacity as the Minimum Diameter Multiple Steiner Tree (MDMST) problem stated below.

Statement 1-1. *Given an undirected edge-weighted graph $G = (V, E)$ with two associated functions **lat**, **cap** where **lat** $: E \mapsto \mathbb{R}^+$ denotes the latency of an edge E and **cap** $: E \mapsto \mathbb{N}^+$ denotes the bandwidth capacity for an edge E, construct k Steiner trees $T_i = (V_{T_i}, E_{T_i})$ in G such that $V_{T_i} \subseteq V$, $E_{T_i} \subseteq E$, $i = 1, \ldots, k$.*

For each Steiner tree, the user should also define a subset of vertices (terminal points) to be included in the tree as well as a minimum required bandwidth.

Statement 1-2. *k subsets of vertices S_i are also defined along with their required minimum bandwidth $bw_i \in \mathbb{N}^+$ for each output Steiner tree T_i such that $S_i \subseteq V_{T_i} \subseteq V$, $i = 1, \ldots, k$*

2.1 Problem Objective

The objective of this problem is to solve the optimization problem in the below statement.

$$\begin{aligned} &\text{minimize} && \textstyle\sum_{i=1}^{k} \mathbf{lat}(T_i) \\ &\text{subject to} && \forall e \in E, \quad \textstyle\sum_{i=1}^{k} bw_i x_i \le \mathbf{cap}(e). \\ & && x_i = \begin{cases} 1, & e \in T_i \\ 0, & e \notin T_i \end{cases} \end{aligned} \tag{1}$$

The latency for each tree is defined as the maximum latency between any two vertices in the subset S_i while the latency between two vertices in the subset S_i is defined as the sum of all the latency lying on the unique path P_{uv} in T_i connecting u and v.

$$\mathbf{lat}(T_i) \triangleq \max_{u,v \in S_i, u \neq v} \mathbf{lat}(u, v) \tag{2}$$

$$\mathbf{lat}(u, v) \triangleq \textstyle\sum_{e \in P_{uv}} \mathbf{lat}(e) \tag{3}$$

Equation 2 is also renown as the diameter of a tree.

1 Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto 606–8501, Japan
2 Academic Center for Computing and Media Studies, Kyoto University, Sakyo-ku, Kyoto 606–8501, Japan
a) lee@net.ist.i.kyoto-u.ac.jp
b) shuichi@media.kyoto-u.ac.jp
c) okabe@i.kyoto-u.ac.jp

3. Method

To solve the defined problem, we divide the process of our method into a loop of two main steps. First, we find the minimum diameter Steiner tree (MDST) T_i for a given S_i provided that the bandwidth capacity of each edge is sufficient enough. Next, we embed the tree T_i into graph G by substracting the minimum required bandwidth bw_i of the constructed tree T_i from the bandwidth capacity of the edges $\mathbf{cap}(e)$ used in the previous step. The operation of these two steps are repeated until the bandwidth capacity is not sufficient enough to construct another tree.

Algorithm 1 MDMST heuristic

```
for i=1,...,k do
    if all pairs u, v ∈ S_i are connected then
        (connected ≜ ∀e ∈ ∃P_uv, bw_i < cap(e))
        find the MDST T_i for S_i
        ∀e ∈ T_i, cap(e) ← cap(e) − bw_i
    else
        terminate loop
    end if
end for
```

3.1 Minimum Diameter Steiner Tree for each given S_i

Hassin and Tamir[2] provided the proof that if x^* is an absolute 1-center of a graph G, then the shortest path tree connecting x^* to all other nodes $v \in V$ is a minimum diameter spanning tree of G. Ding and Qiu[3] generalized the proof such that if x^* is an absolute 1-Steiner center of G, then the shortest Steiner path tree connecting x^* to all other nodes $v \in V$ is a minimum diameter Steiner tree of G. Furthermore, x^* must be at the midpoint of the longest path (diameter) in $T(x^*)$.

In the case of VLAN configuration, since we can only change the settings (access and trunk ports configuration) on each switch (vertices) rather than on the links (edges). Therefore, instead of finding the absolute 1-Steiner center, we aim to find the vertex 1-Steiner center v^* in G.

Definition 1. *A $v^* \in V$ is called the vertex 1-Steiner center if the longest shortest path distance between v^* and $s \in S$ is minimum. $max_{s\in S}\, d(v^*, s) = min_{v\in V} max_{s\in S}\, d(v, s)$.*

To find the vertex 1-Steiner center v^*, we first use the Dijkstra's algorithm[4] to find the shortest path distances of all pairs. The time complexity of this process requires $O(|V|^3)$ or $O(|V||E| + |V|^2 log(|V|))$ by using a min-priority queue[5] during implementation. Next, for each $v \in V$ we calculate the diameter (maximum distance between two nodes in S that passes through v). Finally, we choose the minimum diameter among all the diameters we obtained for each v. The corresponding vertex is the vertex 1-Steiner center in G for the given S_i.

3.2 Embedding Order

Since the bandwidth capacity of the edges in the given graph is limited, the order in how we embed the trees effects the feasibility to embed all trees. We prioritize the trees such that those with the following characteristic would have higher priority.

- larger bandwidth requirement
- more vertices

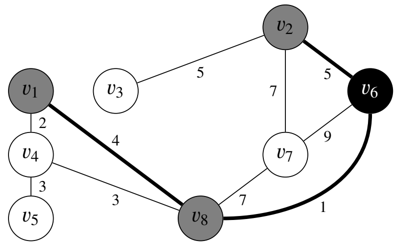

Fig. 1 An example diagram of a MDST constructed from $S = \{v_1, v_2, v_8\}$ (colored in gray). The vertex 1-Steiner center of this graph is v_6 (colored in black). Notice also that the diameter of this tree is the path $\mathbf{lat}(v_1, v_2) = 10$.

4. Evaluation

We perform two evaluations in the following sections, the first one demonstrates the relation between V number and computation time while the second one compares the result between S number and computation time.

Table 1 Average calculation time for constructing a single MDST within a graph of $|V|$ and $|S|$ vertices using the $O(|V|^3)$ all pairs shortest path algorithm. The input adjacency matrix of latency cost is randomly generated with a probability $p = 0.6$ of existing a connection between two vertices in V.

$\lvert V\rvert$ vertices	$\lvert S\rvert$ vertices	average time elapsed (s)
50	10	0.168273
100	20	1.203410
150	30	3.824009
300	60	30.145795
600	120	244.710218

Table 2 Average calculation time for constructing a single MDST with fixed $|V|$ and varying $|S|$ vertices.

$\lvert V\rvert$ vertices	$\lvert S\rvert$ vertices	average time elapsed (s)
150	15	3.880055
150	30	3.824009
150	50	3.858702

It is worth noticing that once the calculation of the shortest path distances of all pairs are done, the remaining calculation time to find the MDST for any given S is relatively short.

5. Concluding Remarks

We construct the MDMST problem in order to solve the multiple VLANs embedding problem as well as introducing a heuristic procedure to obtain a solution. In the case of unlimited bandwidth, the problem is reduced to solve k MDST problems, however when the constraint of the bandwidth capacity is included into the problem, cases in which it is not possible to embed all MDSTs will exist. We will research these cases in future works.

References

[1] "802.1 Q/D10, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks, Copyright by the Institute of Electrical and Electronics Engineers," 1997.

[2] R. Hassin and A. Tamir, "On the minimum diameter spanning tree problem," *Information processing letters*, vol. 53, no. 2, pp. 109–111, 1995.

[3] W. Ding and K. Qiu, "Algorithms for the minimum diameter terminal steiner tree problem," *Journal of Combinatorial Optimization*, pp. 1–17, 2013.

[4] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[5] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the ACM (JACM)*, vol. 34, no. 3, pp. 596–615, 1987.